Lab Assignment – 8

Please follow the following document for writing code for Lab Assignment 8 (BST Operations)

1. Class: TreeNode

Constructor:

- `__init__(self, val: int) -> None`
  - Parameters:
    - `val` (int): The value of the node.
  - Description:
    - Initializes a new TreeNode with a given integer value.
    - Sets left and right child nodes to `None`.

2. Class: BinarySearchTree

Constructor:

- `__init__(self) -> None`

  ```
  self.root = None
  ```

  - Parameters:
    - None
  - Description:
    - Initializes an empty binary search tree with `root` set to `None`.

Methods:

- `insert(self, val: int) -> None`
  - Parameters:
    - `val` (int): The value to be inserted.
  - Description:
    - Inserts a new node with the given value into the BST while maintaining BST properties.
- `search(self, val: int) -> bool`
  - Parameters:
    - `val` (int): The value to search for.
  - Returns:
    - `True` if the value is found, otherwise `False`.
  - Description:
    - Searches for a node with the specified value in the BST.
- `delete(self, val: int) -> None`
  - Parameters:

        - `val` (int): The value to delete.
    - Description:
        - Deletes a node with the specified value and adjusts the tree accordingly.
- `inorder_traversal(self) -> list[int]`
    - Returns:
        - A list of integers representing an in-order traversal of the BST.
    - Description:
        - Performs an in-order traversal (left-root-right) and returns the values in sorted order.
- `preorder_traversal(self) -> list[int]`
    - Returns:
        - A list of integers representing a pre-order traversal of the BST.
    - Description:
        - Performs a pre-order traversal (root-left-right) and returns the values.
- `postorder_traversal(self) -> list[int]`
    - Returns:
        - A list of integers representing a post-order traversal of the BST.
    - Description:
        - Performs a post-order traversal (left-right-root) and returns the values.

Usage Example:

```
bst = BinarySearchTree()
bst.insert(20)
bst.insert(15)
bst.insert(17)
bst.insert(10)
bst.insert(11)
bst.insert(13)
bst.insert(12)
print(bst.preorder_traversal())
```

This document outlines the structure and functionality of the `BinarySearchTree` and `TreeNode` classes, detailing all methods, parameters, return types, and their roles.